

7. プログラムの実行順序

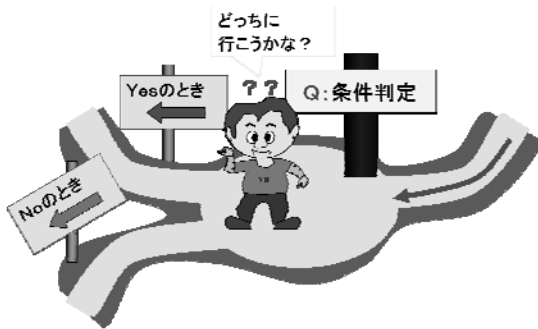
(1) 条件分岐

プログラムでは、基本的にソースプログラムの前から後の方向に実行します。

```
int A, B, C;  
A = 10;  
B = 20;  
C = A + B;  
printf("C = %d", C);
```

実行順序

ただし、条件によっては、プログラムの流れを変える必要があります。したがって、条件判定が必要です。



(2) 条件判定の方法

条件判定は、以下の2つの方法、あるいはこれらの組合せで行います。

- 関係式
- 論理式

関係式は、以下のように書きます。

● A は B より大きいか?	A > B
● A は B より小さいか?	A < B
● A は B 以上か?	A >= B
● A は B 以下か?	A <= B
● A は B と等しいか?	A == B
● A は B が等しくないか?	A != B

演算子「>=」、「<=」、「!=」は、ASCII に「≥」、「≤」、「≠」の文字がないため、このように書くんだと解釈してください。

演算子「==」は、代入文の「=」と区別するためにこう書きます。

これらは通常の式の記号と異なりますが、約束事ですので慣れましょうね。

これらの式では、

- 結果が真でないとき0
- 真のときゼロでない値(非ゼロ)

になります。なお、非ゼロを定数として内部的にどう表現するか(1あるいは-1、ANSI 標準では1)は、処理系によって異なります。

これを確認するには、以下のようなプログラムを作って実行してみると良いでしょう。

```
#include "stdafx.h"  
int main()  
{ int a; a = 10 ==10;  
  printf("¥n *** 10 ==10 : %d", a);  
}
```

ただし、真のときの値が1になっているからといって、論理値以外の意味で使うのはお勧めできません。あくまで論理値として扱きましょう。たとえば、以下のような表現でAに0の値を期待するような記述は、もってのほかです。

$$A = (B == C) - 1;$$

同様に、偽のときの値が0になっているからといって、以下の記述で、Aに1の値を期待するようなこともやめましょう。